WHITEPAPER · v1.3 · MAY 2026

Agentic Commerce Intelligence & Risk on Stablecoin Settlements

A reference architecture for measuring, attributing, and protecting agent-initiated checkout flows that settle in fiat rails and on-chain stablecoins.

Authors	Corgi Labs Research
Version	1.3 — May 12, 2026
Audience	Payments, risk, and platform engineering teams
Companion	corgi-agent-payments-dashboard.vercel.app

Abstract. Agent platforms now initiate a meaningful share of online checkout volume, and a growing slice of that volume settles in stablecoins instead of card rails. This paper defines a unified detection, attribution, and risk-scoring model that spans both worlds: card-based agent checkout reviewed by Stripe and stablecoin transfers settled on Base, Solana, and Ethereum L1. We describe the signal classes Corgi uses at the request edge, the funnel attribution graph that joins on-chain confirmations to off-chain intent, and the benchmark priors we publish across major agent platforms.

Contents

01 · Why agentic commerce needs its own measurement layer

Conventional payments analytics is built around a session and a device. A human opens a tab, a fingerprint is collected, a card is entered, and the resulting authorization is attributed back to that session. Agent-initiated checkout violates every assumption in that pipeline.

Agents operate as autonomous browsers and tool chains. They reuse credentials across many merchants, recover from failure with deterministic retries, and frequently settle in stablecoins through programmable wallets rather than personal cards. Treating that traffic as anonymous human sessions produces three concrete failures: under-counted volume, misattributed declines, and inflated dispute and fraud rates that punish the wrong cohort.

Three failure modes we observe in the wild

- **Volume blindness.** 18–34% of checkout attempts on the merchants we sample originate from agents, but appear in standard analytics as direct or referral traffic.
- **Decline misattribution.** Issuer declines triggered by agent-shape signals are reported as generic *do_not_honor*, masking a recoverable failure mode.
- **Settlement opacity.** Stablecoin settlements bypass the card processor entirely, leaving finance teams with no unified view of total agent GMV.

02 · Detection: a four-signal model at the request edge

Corgi's detection layer runs as an edge worker in front of the merchant's checkout. Each incoming request is scored on four signal classes, combined into a calibrated probability that the request was initiated by an agent. Added latency is held under 8 ms p95.

Signal class	What it captures	Example features
Behavioral	Human vs. machine timing	Keystroke cadence, mouse entropy, page-time distribution
Network	Origin and routing shape	Known agent egress IPs, ASN reputation, request fan-out
Fingerprint	Client identity	Tool-specific user agents, JA4/JA3 hashes, client hints
Session shape	Topology of the attempt	Idempotency reuse, retry tree, intent-to-checkout latency

Each class produces a partial score; the combiner is a calibrated logistic model retrained weekly against labeled traces from partner merchants. Calibration is verified with an expected calibration error

(ECE) target below 0.02 across deciles.

03 · Attribution across off-chain and on-chain settlement

Detection alone is not measurement. To produce useful metrics we stitch each detected attempt to its terminal state — through Stripe's risk review and the issuer authorization decision for card flows, and through the on-chain confirmation and finality window for stablecoin flows. The unifying object is a *trace*: a single record that joins the request edge to the settlement event, regardless of rail.

Trace schema (abbreviated)

```
trace_id, agent_score, platform_hint, intent_hash, rail (card | usdc-base | usdc-sol |
usdt-eth),
stripe_payment_intent_id?, issuer_decision?, on_chain_tx?, finality_block?,
settled_amount,
outcome (authorized | declined | flagged | confirmed | reorged | refunded), latency_ms
```

The same trace primitive carries both rails. A card attempt reaches its terminal state when the issuer responds; a stablecoin attempt reaches it when the relevant finality window has passed without reorg. This lets a finance team report a single agent GMV figure that sums card and stablecoin settlement, and lets a risk team compare auth-failure rates against on-chain failure modes (insufficient balance, allowance not granted, slippage rejection) with consistent denominators.

04 · Stablecoin settlement: rails, finality, and reversal risk

Stablecoin settlement removes the issuer from the loop but introduces a different risk surface. Corgi treats each rail as a first-class settlement channel with its own finality model, refund mechanics, and observable reversal patterns.

Rail	Typical finality	Reversal model	Observed agent share
USDC on Base	~2 s soft, 12 min hard	Dispute via merchant refund tx; no chargeback	31%
USDC on Solana	~0.4 s, single-slot	Refund tx only; high MEV-style griefing	22%
USDT on Ethereum L1	~12 s per block, 6-block confidence	Refund tx; permit/permit2 race conditions	9%
Card (Visa/MC) via Stripe	Auth in ms, settlement T+2	Chargeback up to 120 days	38%

Source: Corgi aggregate, rolling 90 days ending May 2026.

Reversal-risk priors

Stablecoins have no chargeback, but they are not risk-free for merchants. We track three reversal-adjacent failure modes that drive economic loss without the issuer's involvement:

- **Reorg exposure.** Goods released before hard finality can be lost on minority reorgs. Corgi tags every settlement with the finality window applied at release time.
- **Refund-tx abuse.** Agents that successfully request a refund and then dispute the original card leg (in mixed-rail flows) generate a synthetic chargeback path.
- **Permit and allowance griefing.** EIP-2612 / Permit2 flows are vulnerable to front-running that strands signed approvals; the merchant sees a confirmed signature and no transfer.

05 · Risk scoring and the Corgi priors

For each trace we emit a risk score in [0, 1] that is rail-aware: the same agent_score maps to different recommended actions on a 12-minute Base finality window than it does on a Stripe authorization with chargeback exposure. The score is consumed as a header on the merchant's existing checkout response, or as a webhook for asynchronous flows.

Recommended actions by score band

Band	Card rail	Stablecoin rail
0.00 – 0.30	Allow	Allow, release at soft finality
0.30 – 0.60	Allow + 3DS step-up	Hold until hard finality
0.60 – 0.85	Review queue	Hold + manual review
0.85 – 1.00	Decline	Decline before signature

06 · Reference integration

A typical merchant deploys Corgi in two places: an edge middleware in front of /checkout, and a settlement listener that reconciles Stripe webhooks together with on-chain Transfer events on the merchant treasury wallet.

Edge middleware (pseudocode)

```
const verdict = await corgi.score(request);
if (verdict.action === 'decline') return new Response('blocked', { status: 402 });
request.headers.set('x-corgi-trace', verdict.trace_id);
request.headers.set('x-corgi-score', verdict.score.toFixed(3));
return fetch(origin, request);
```

Settlement listener

```
corgi.on('stripe.payment_intent.succeeded', joinTrace);
corgi.on('chain.transfer', (ev) => joinTrace({
rail: ev.chain, tx: ev.hash, amount: ev.value, finality: ev.confirmations
}));
```

07 · Benchmarks & appendix

Aggregated across the merchants in the Corgi panel, the following priors describe agent checkout health for the 90-day window ending May 2026. Use them as a sanity baseline; per-merchant baselines are produced inside the dashboard.

Metric	Card (agent)	Card (human)	Stablecoin (agent)
Approval rate	84.1%	92.6%	97.8%
Auth rate	78.9%	91.2%	—
Payment success rate	71.4%	88.0%	95.2%
Abandonment rate	18.0%	9.4%	3.1%
Dispute rate	1.4%	0.6%	0.2% (refund-tx)
Confirmed fraud rate	0.42%	0.18%	0.09%

n = 184 merchants, 312M attempts, 9.4M confirmed agent traces. Stablecoin sample is biased toward USDC on Base (61% of on-chain settlement volume).

Open questions

- Should reorg-window risk be priced into merchant-of-record fees, the way card interchange embeds chargeback risk?
- How do we standardize a cross-rail dispute primitive so finance teams can reconcile mixed-rail refunds without bespoke pipelines?
- What is the right disclosure surface to end users when an agent settles on their behalf in a stablecoin rather than the card on file?